Global Scientific JOURNALS

GSJ: Volume 12, Issue 5, May 2024, Online: ISSN 2320-9186
www.globalscientificjournal.com

A COMPARATIVE STUDY ON INDEXING AND PERFORMANCE OF SQL SERVER AND MYSQL

Mr. Suresh Kumar, Lecturer in Department of Information Technology, University of Technology and Applied Science, Salalah, Sultanate of Oman

Dr. ChandraPal, Lecturer in Department of Information Technology, University of Technology and Applied Science, Salalah, Sultanate of Oman

ABSTRACT

A database is collection of interrelated data that is well organized, typically stored electronically so that it can be easily accessed, modified and managed. There are different types of Databases e.g. Relational. Object-oriented, NoSQL databases, Graph databases etc. The choice of database depends on nature of data, volume of data and how it is intended to be used e.g., Relational databases are the best to handle structured data whereas NoSQL databases are best suited for handling large amount of unstructured-semi-structured data.

A database is usually controlled by an application software called database management system (DBMS). DBMS serves as an interface between the database and its end users or programs, allowing users to retrieve, update, and manage how the information is organized and optimized. A DBMS also facilitates oversight and control of databases, enabling a variety of administrative operations such as performance monitoring, tuning, and backup and recovery. Some examples of popular database software or DBMSs include MySQL, Microsoft Access, Microsoft SQL Server, FileMaker Pro, Oracle Database and MongoDB.

As per Statista[1], as of September 2023, the most popular relational database management system (RDBMS) in the world was Oracle, with a ranking score of 1240.88. Oracle was also the most popular DBMS overall. MySQL and Microsoft SQL server rounded out the top three. One of the parameter to measure performance of a database is response time for the queries that are submitted by the user either directly using SQL or through GUI based front-end applications. Indexing is one of the techniques using to improve query response time. Finally, the paper presents a comparative study

of the Performance of MySQL and Microsoft SQL server based on test cases that will include different cases to compare the response time performance of indexes in both these databases.

KEYWORDS

Indexing, Relational database performance comparisons, MySQL, Microsoft SQL server

1. INTRODUCTION

Most of the OLTP-data driven applications use relational databases to store and manage large amount of data. Overall performance of such applications depends on database performance. Therefore, database performance optimization is an absolute must for the success of these applications.

To optimise database performance, there are different strategies that must be considered. Some of them includes[2] deploying a third party performance solutions, using the latest versions of the database for specific operating systems, considering business requirements for selecting different editions of a specific DBMS, choosing the most optimal data types for fields in tables, concurrency control methods to avoid locking and deadlocks, optimizing queries and optimising indexing etc.

Considering the fact that best suited DBMS edition for critical business requirements have been deployed on latest operating system platform, optimizing Indexing is considered core performance tuning method. SQL queries must be well constructed for getting benefit of suitable indexing.

There are different types of indexing that are supported by relational databases. SQL Server support[3] Clustered, Non-clustered, unique and filtered index types mainly. On the other hand, main MySQL indexes include[4] PRIMARY KEY, UNIQUE, and INDEX. In additions to these index types, there are additional index types in both the databases that are used in special cases for special types of data e.g., spatial index is used for SPATIAL data only and so on.

Here are some common types of indexing in both SQL Server and MySQL:

Single-Column Index: This is the most basic type of index, where an index is created on a single column of a table. It speeds up queries that filter or sort data based on that column.

Composite Index: Also known as a multi-column index, this type of index involves creating an index on multiple columns. Composite indexes can be beneficial when queries involve filtering, sorting, or joining based on multiple columns.

Unique Index: A unique index ensures that the indexed columns contain only unique values, similar to a primary key constraint. It prevents duplicate values from being inserted into the indexed columns.

Clustered Index (SQL Server) / Primary Key (MySQL): In SQL Server, a clustered index determines the physical order of data in a table. In MySQL, the primary key constraint serves a similar purpose. Each table can have only one clustered index or primary key.

Non-clustered Index (SQL Server) / Index (MySQL): In SQL Server, a non-clustered index is a separate structure from the table and contains pointers to the actual data rows. In MySQL, an index is similar to a non-clustered index in SQL Server. Multiple non-clustered indexes can be created on a table.

Full-Text Index (MySQL): Full-text indexing enables efficient searching of text data stored in large text columns such as VARCHAR or TEXT. It allows for complex searches involving keywords, phrases, and proximity operators.

Spatial Index (MySQL): Spatial indexing is used to optimize queries involving spatial data types such as points, lines, and polygons. It accelerates operations like nearest neighbour searches and spatial joins.

Hash Index (MySQL): Hash indexing is a technique used to quickly locate a single row based on the hash value of one or more columns. It's particularly useful for equality searches but doesn't support range queries.

These are some of the common types of indexing available in SQL Server and MySQL. Choosing the appropriate type of index depends on factors such as the structure of your data, the types of queries you frequently run, and the performance requirements of your application.

The major type of indexes in both can be identified as below:

S. No.	SQL Server	MySQL	Remarks
1	Clustered	PRIMARY KEY	PRIPARY KEY in MySQL is similar to Unique clustered index in SQL Server
2	Non-clustered	INDEX	Also called simple, regular, or normal or secondary index in MySQL
3	Unique	Unique	Same meaning in both the databases but in MySQL if there is no Primary key then it work as Clustered Index

Thus, the main index type is Clustered index which is implemented through primary key in most of the situations. Implementation in both the databases is different.

2. PROBLEM DEFINITION

MySQL and SQL Server both are widely used Relational Databases. Normally used indexes are clustered, primary keys, non-clustered and unique. Response time is very critical for data driven applications. In this paper focus will be on comparing the performance of both these databases in the light of these indexes using different tables with different number of indexes. We want to check which implementation is faster for the same sample database.

3. METHODLOGY

There are many sample relational free Open-Source Test Databases[5] that can be used for Testing. The sample database that was used Chinook, a **medium-size** database. We have downloaded the SQL scripts shared from GitHub repositories to set up testing environment. Index on a table affect all the operations that can be performed on a table.

Test cases focus on study of effect of Index on these operations and compare performance of MySQL and SQL Server in each case. The following test cases are performed:

Case-1: Queries involving one table (master table) without any foreign keys (Artist): In case-1, we have only one index which is Primary key

Case-2: Queries involving two tables (master/detail) with one foreign key (Album, artisist): In this case Album has FK to Artist table and hence we have two indexes in this case.

Case-3: Queries involving more than two tables (track, albums, and Genre): Here, Track table is linked to Album, Genre and some more tables through foreign keys. Therefore, queries involving these tables will involve at least three indexes.

These three types of cases have been used in INSERT, UPDATE, DELETE and SELECT queries. The execution time has been recorded to study the effect of Indexes in each cases.

The queries have been executed on standalone Databases (SQL-Server 2022 and MySQL: 8.0.31 - MySQL Community Server – GPL) and the results have been recorded in table in the next section.

4. RESULTS

The response times for all these different cases have been summarized in the table below:

CPU Time taken in seconds

	Case-1		Case-2		Case-3		Case-4*	
	SQL	MySQL	SQL	MySQL	SQL	MySQL	SQL	MySQL
Insert	.0000010	.0005	.0001601	.0003	.0004833	.0010	-	-

Update	.0000010	.0019	.0000010	.0010	.0000010	.0009	0.0001000	.0015
Delete	.0001634	.0003	.0001736	.0002	.0002736	.0014	-	-
SELECT	.0004606	.0003	.00126292	0.0020	.01381976	.0004		

*operation involves CHANGE in foreign key

5. PERFORMACE COMPARISIONS

Let's compare the results recorded in last section.

Case-1: Single table and single index:

Operation	SQL	MySQL	Faster
Insert	.0000010	.0005	SQL
Update	.0000010	.0019	SQL
Delete	.0001634	.0003	SQL
SELECT	.0004606	.0003	MySQL

Case-2: Two tables with Join and two indexes:

Operation	SQL	MySQL	Faster
Insert	.0001601	.0003	SQL
Update	.0000010	.0010	SQL
Delete	.0001736	.0002	SQL
SELECT	.00126292	0.0020	MySQL

Case-3: Three tables with Join and three indexes:

Operation	SQL	MySQL	Faster
Insert	.0004833	.0010	SQL
Update	.0000010	.0009	SQL
Delete	.0002736	.0014	SQL
SELECT	.01381976	.0004	MySQL

It is clear that the clustered index in SQL server and is performing far better than the Primary key in MySQL. But when it comes to SELECT queries, MySQL is performing better than the SQL-Server. But it must be noted that the volume of data in sample database **Chinook**, a **medium-size** database is not very large. As the column of data increases the results will change drastically.

Small to medium sized databases, that require more INSERT, Update, Delete operations, SQL is go ahead choice for database. However, if the database is more read-only nature but data volume is medium such as used in Warehouse MySQL server may perform better. This shows that updating indexes is costly in case of MySQL as compared to SQL Server.

6. CONCLUSION

In this paper, we have presented main type of index used in SQL-Server and MySQL database. SQL Server offers advanced indexing options such as filtered indexes, which allow you to create indexes based on a subset of rows that meet specific criteria. Additionally, SQL Server supports XML indexes for efficient querying of XML data, which MySQL does not natively support. SQL Server provides specialized indexing features such as column store indexes for optimizing analytical processing (OLAP) workloads. Column store indexes store and query data by column rather than by row, resulting in improved query performance for analytics queries involving large datasets. While MySQL has introduced similar features in recent versions, SQL Server has a longer history of optimizing for OLAP workloads

In terms of performance comparison on similar indexes, clustered index in SQL Server and Primary key index in MySQL, SQL-Server is better and efficient database in terms of CRUD operations and hence suited to all CURD sensitive OLTP systems. On the other hand Index implementation in MySQL is faster for SELECT queries. Therefore, application that have medium size and requires mostly read operations such as analysing data, MySQL may be preferred.

REFERENCES

1. https://www.statista.com/statistics/1131568/worldwide-popularity-ranking-relational-database-management-systems/
2. https://www.sentryone.com/blog/10-steps-to-better-database-performance
3. https://learn.microsoft.com/en-us/sql/relational-databases/sql-server-index-design-guide?view=sql-server-ver16
4. https://dev.mysql.com/doc/refman/8.0/en/mysql-indexes.html
5. https://data-xtractor.com/blog/data-modeling/test-databases/